

DCA0214.1 - LABORATÓRIO DE ESTRUTURAS DE DADOS

Aula 10: Listas de prioridades e Heaps

Prof. Felipe Fernandes

10 Maio de 2019

— LEIA COM ATENÇÃO —

- I É proibido utilizar qualquer estrutura de dados ou algoritmos pré-fornecidos por alguma biblioteca C/C++.
 - II Em cada aula desta unidade, as listas de exercícios possuirão questões marcadas com \star . Todas as questões (de todas as aulas) marcadas com \star , devem ser submetidas via SIGAA até às 23h59 do 16 Maio 2019. Peso: 20% na nota de segunda unidade.
 - III O item II não lhe desobriga de resolver as questões sem \star . Todas as questões são importantes para a prova da segunda unidade.
-

- 1. Escreva uma função que recebe um um vetor $V[1...n]$ e decide se V é um *heap max* ou *min* ou nenhuma dessas possibilidades.
- 2. Implemente as seguintes operações básicas para *heap max*:
 - (a) Subir (iterativo e recursivo)
 - (b) Descer (iterativo e recursivo)
 - (c) Construção
 - (d) Remoção
 - (e) Selecionar o de maior prioridade
- 3. \star Dados um vetor $V[1...n]$ de inteiros positivos **desordenados** e dois inteiros positivos x e k , implemente um procedimento que retorna os k valores em V mais próximo de x . Exemplo: $V = \{10, 2, 14, 4, 7, 6\}, x = 5, k = 3$. Os três valores mais próximos de 5 são: 4, 6 e 7. Sua implementação deve ter complexidade $O(n \log k)$.

4. Utilizando uma *heap max*, implemente o algoritmo de ordenação *HeapSort*.
5. ★ Um ladrão invadiu um cofre de banco, onde $n \geq 1$ barras de ouro estão guardadas. As barras de ouro estão numeradas de 1 a n . Cada barra $i \in \{1, \dots, n\}$ possui uma massa, em *kg*, denotada por w_i , e possui também um valor equivalente em reais, denotado por v_i . O ladrão possui uma mochila, na qual ele pretende transportar as barras de ouro roubadas. A mochila do ladrão pode transportar uma carga (massa) máxima de W quilogramas. Suponha que o ladrão possui um mecanismo de quebrar uma barra de ouro a fim de, caso necessário, fracioná-la e facilitar seu transporte. Massa e valor são igualmente distribuídos por cada fração de uma barra. Por exemplo, se uma barra i tem $v_i = R\$120$ e $w_i = 30kg$, então $\frac{2}{3}$ da barra i valem $v_i = R\$80$ e $w_i = 20kg$. Sua tarefa é ajudar o ladrão a escolher um subconjunto de barras de ouro a serem roubadas de modo a maximizar o valor em reais obtido na ação, respeitando o peso carregado na mochila do meliante. Considere que o ladrão pode entrar no cofre do banco apenas uma vez (ou seja, ele não pode regressar). Implemente um algoritmo que recebe w_i , v_i , $i \in 1, \dots, n$, e W e retorna o valor (máximo) do roubo. Complexidade requerida no pior caso: $O(n \log n)$. OBS.: você não pode utilizar nenhum algoritmo de ordenação.

